

FIG. 1

default.asp

210

302

```
<% Option Explicit %>
<!--#include file="../../tasks/tasks.asp"-->
<!--#include file="../../utils/xmlutils.asp"-->
<!--#include file="../../utils/sysutils.asp"-->
<!--#include file="../../utils/dbutils.asp"-->
<%
dim objXML, reqPage, goPage, xslfile, x, locale

'Get the locale name which is the last directory name in the current path without the "\".
'This id can be used later in any tasks that have locale specific data.
locale = mid(server.MapPath("."), len(server.MapPath("../")) + 2)

'load the master xml structure into a dom object - will be used globally
set objXML = getXMLDoc (Server.MapPath("../xml/master.xml")) 304

'append the images and labels xml structures for the locale to master.xml
AppendXML(Server.MapPath("xml/images.xml"))
AppendXML(Server.MapPath("xml/labels.xml")) 306

'get the page value off the form if it exists
reqPage = Request("page")

'if reqPage exists then loop the tasks for building the requested page otherwise default tp page 1
if reqPage < ">" then
    goPage = RunTasks(reqPage)
    if goPage < reqPage then it will be treated like a redirect and the new goPage will need to
    have its tasks run. this will only be done 1 time.
    if goPage < reqPage then
        goPage = RunTasks(goPage)
    end if
else
    goPage = RunTasks("1")
end if

'Append the variable x xml structure into the objXML structure for use in the xsl templates.
'This will only be run if some values have been populated into the variable.
if x < ">" then
    x="<builddata>" & x
    'the task xml nodes that will be built indside here are in the tasks.asp - that is why x is a
    global variable
    x=x & "</builddata>"
    AppendXML(x)
end if

'build the page with the XML structure loaded and appended to above using the XSL file for the next page
number
xslfile = server.MapPath("../xsl/page" & goPage & ".xsl") 308
Response.write TransformXML(objXML, xslfile )

'cleanup the objects on this page - all others are cleaned on the individual pages
set objXML = nothing
%>
```

FIG. 3

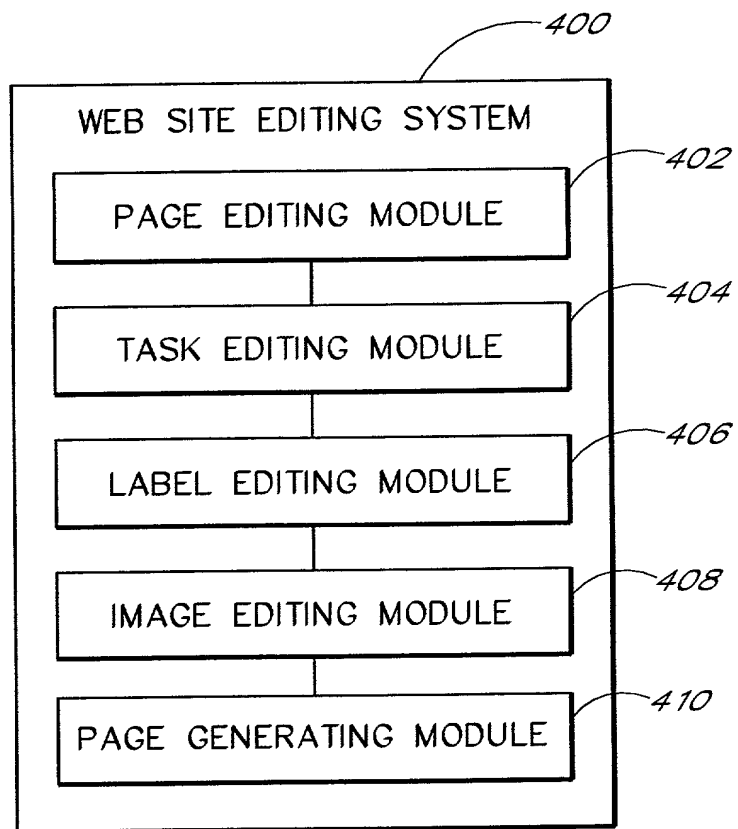


FIG. 4

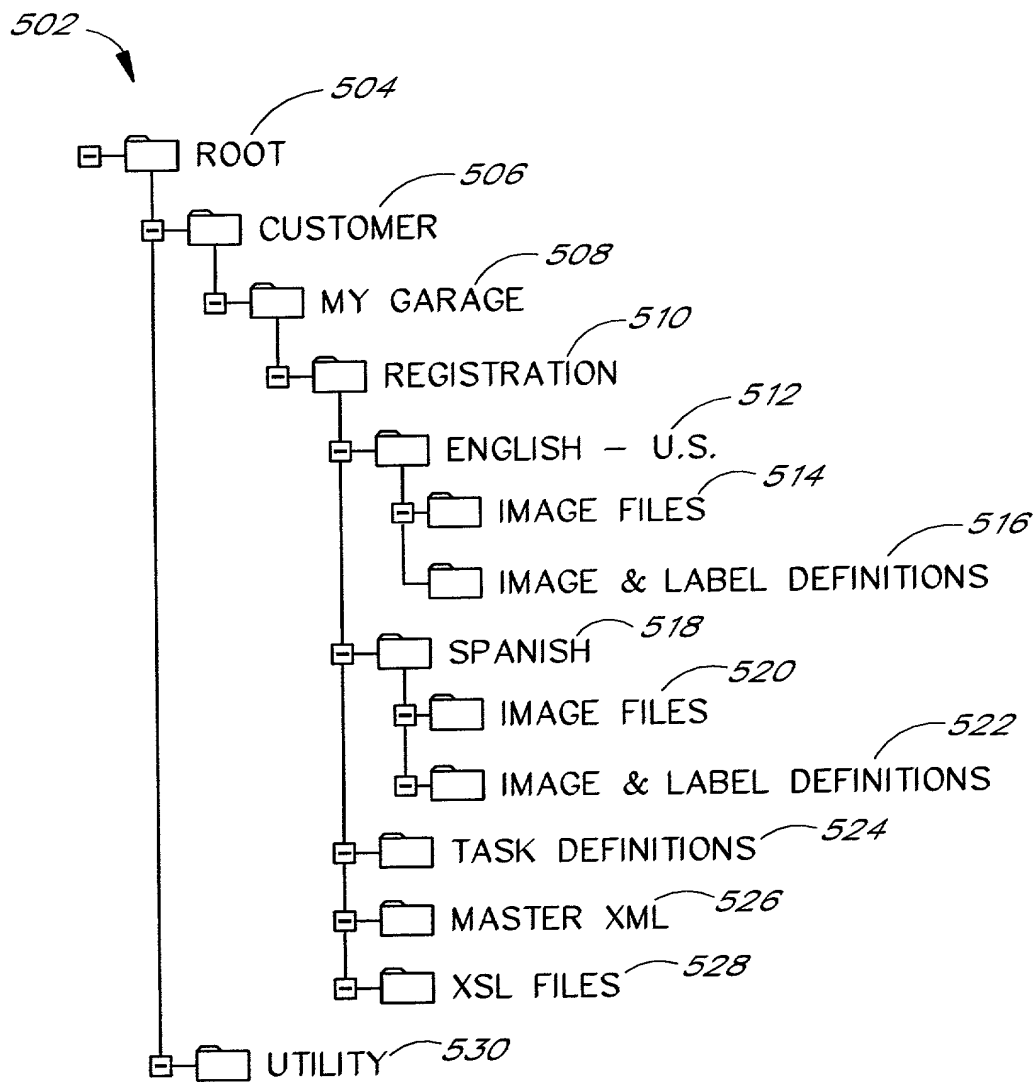


FIG. 5

```

602  <?xml version="1.0"?>
604  <-process product="Customer" program="My Garage" process="Register">
    <-tasks>
        606 {
            <task id="1" name="validateEmailAddress"/>
            <task id="2" name="isEmailAddressAvailable"/>
            <task id="3" name="validateUserInformation"/>
            <task id="4" name="createUserInformation"/>
            <task id="5" name="getUserInformation"/>
            <task id="6" name="lookforemail"/>
        }
    </tasks>
    <-pages>
        610 <-page id="1" name="Enter email address">
            612 <build/>
        </page>
        610 <-page id="2" name="Re-enter email address">
            612 <-build>
            <task id="1" success="page 1" fail="page 1" />
            </build>
        </page>
        610 <-page id="3" name="Email exists">
            612 <-build>
            <task id="5" success="page 3" fail="page 7" />
            </build>
        </page>
        <-page id="4" name="Enter user information">
            <-build>
            <task id="1" success="page 1" fail="page 1" />
            </build>
        </page>
        <-page id="5" name="Re-enter user information">
            <build/>
        </page>
        <-page id="6" name="info entered">
            <-build/>
            <task id="3" success="task 4" fail="page 5"/>
            <task id="4" success="page 6" fail="page 7"/>
            </build>
        </page>
        <-page id="7" name="System error message">
            <build/>
        </page>
        <-page id="8" name="Logon Page">
            <build/>
        </page>
    </pages>
</process>

```

608

FIG. 6

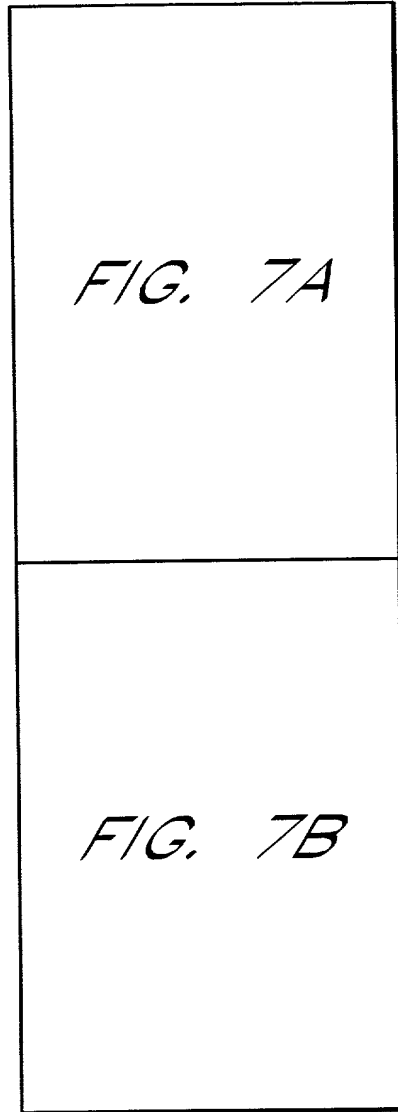


FIG. 7

```

function validateEmailAddress()
    dim email
    'Get the email address variable
    email = Request("email")

    if email <> "" then
        'check to make sure the email address is properly formed
        if instr(1, email, "@") > 0 then
            validateEmailAddress = "Pass"
        else
            'the next xsl page will need the email address in the xml structure
            BuildElementXML "emailaddress", email
            validateEmailAddress = "Fail"
        end if
    else
        'the next xsl page will need the email address in the xml structure
        BuildElementXML "emailaddress", email
        validateEmailAddress = "Fail"
    end if
end function

```

```

function isEmailAddressAvailable()
    dim spCMD, rsSP, email

    'Get the email address variable
    email = Request("email")

    Set spCMD = SetSPCMD("GetEmailAddress", strConnect) 'strConnect is set in the
connect.asp include file
    spRet spCMD 'A returned value will occur
    spIntIn spCMD, "@MyAge", UserAge
    spVarCharIn spCMD, "@emailAddress", 255, email '255 character email address
    set rsSP = spCMD.execute 'execute the stored procedure

    'stored procedure result of success or fail
    if spCMD.parameters("RETURN_VALUE") then
        isEmailAddressAvailable = "Fail"
    else
        'task completed ok
        isEmailAddressAvailable = "Pass"
    end if

    'Regardless of the outcome - the email address is needed on the next page.
    BuildElementXML "emailaddress", email
    set spCMD = nothing
    set rsSP = nothing
end function

```

```

function validateUserInformation()
    validateUserInformation = "Pass"
end function

```

```

function createUserInformation()
    dim spCMD, rsSP
    'form element variables
    dim email, fname, lname

    'Get the email address, fname and lname variables
    email = Request("email")
    fname = Request("fname")
    lname = Request("lname")

```

FIG. 7A


```

        Set spCMD = SetSPCMD("CreateUserInformation", strConnect)
spRet spCMD 'A returned value will occur
'spIntIn spCMD, "@MyAge", UserAge
spVarCharIn spCMD, "@emailAddress", 255, email '255 character email address
'need to make a text input parameter or adLongVarChar
dim t
t = "<data name="" + "fname" + "">" & fname & "</data>"
t = t + "<data name="" + "lname" + "">" & lname & "</data>"

spVarCharIn spCMD, "AXMLElements", len(t), t 'xml structure

set rsSP = spCMD.execute 'execute the stored procedure

'stored procedure result of success or fail
if spCMD.parameters("RETURN_VALUE") then
    createUserInformation = "Fail"
else
    createUserInformation = "Pass"
    'add records to xml??
end if

set spCMD = nothing
set rsSP = nothing
end function

```

```

function getUserInformation() 710
    dim spCMD, rsSP, email
    email = Request("email")

```

```

        Set spCMD = SetSPCMD("getUserInformation", strConnect)
spRet spCMD 'A returned value will occur
'spIntIn spCMD, "@MyAge", UserAge
spVarCharIn spCMD, "@MyAge", UserAge
spVarCharIn spCMD, "@emailAddress", 255, email '255 character email address
set rsSP = spCMD.execute 'execute the stored procedure

```

```

'stored procedure result of success or fail
if spCMD.parameters("RETURN_VALUE") then
    getUserInformation = "Fail"
else
    'get the value of the return record
    x=x + "<task name="" + "getUserInformation" + "">"
    x=x + "<record>"
    x=x + rsSP.fields("XMLElements").value
    x=x + "</record>"
    x=x + "</task>"

```

```

        get UserInformation = "Pass"
end if

```

```

set spCMD = nothing
set rsSP = nothing

```

'Because stored procedures may return multiple recordsets, we need to handle them. Assuming we wish to display all of the recordsets as HTML, the following code suffices:

```

'Do until rsSP Is Nothing
    RS2TABLE rsSP
    rsSP=rsSP.NextRecordSet
'Loop
end function

```

```

%>
<%
function lookforemail() 712
    'lookforemail="Fail"
    lookforemail="Pass"
end function
%>

```

FIG. 7B

```

    <?xml version="1.0" ?>
802 -<labels>
804 -<label id="title">
    <![CDATA[Autobytel.com Customer Registration]]>
802 -</label>
804 -<label id="footer">
    <![CDATA[1997-201 autobytel.com]]>
802 -</label>
804 -<label id="emailAddress">
    <![CDATA[Email address: ]]>
    </label>
    -<label id="emailAddressEnter">
      <![CDATA[Enter your email address ]]>
    </label>
    -<label id="emailAddressFail">
      <![CDATA[Error: '{emailAddress}' is not a valid email address. ]]>
    </label>
    -<label id="emailAddressReEnter">
      <![CDATA[Please re-enter your email address. ]]>
    </label>
    -<label id="infoTitle">
      <![CDATA[Your Title: ]]>
    </label>
    -<label id="infoStreet">
      <![CDATA[Street Address: ]]>
    </label>
    -<label id="infoCity">
      <![CDATA[City: ]]>
    </label>
    -<label id="infoState">
      <![CDATA[State: ]]>
    </label>
    -<label id="enterAgain">
      <![CDATA[Please correct your information and submit again. ]]>
    </label>
    -<label id="InfoEntered">
      <![CDATA[Your information has been successfully entered into the database. ]]>
    </label>
    -<label id="ErrorPage">
      <![CDATA[One of the tasks failed when trying to build the page. This is the
        system error page. ]]>
    </label>
    -<label id="LogonPage">
      <![CDATA[This will be the logon page. ]]>
    </label>
    -<label id="emailBelongsTo">
      <![CDATA[...in development... ]]>
    </label>
    -<label id="NewLabelName">
      <![CDATA[Enter Label Text Here ]]>
    </label>
  </labels>

```

FIG. 8

```

    <?xml version="1.0" ?>
902 -<labels>
904 -<label id="title">
    <![CDATA[Autobytel.com Cliente Registration]]>
902 -</label>
904 -<label id="footer">
    <![CDATA[1997-201 autobytel.com]]>
902 -</label>
904 -<label id="emailAddress">
    <![CDATA[Email se dirigen:]]>
    </label>
    -<label id="emailAddressEnter">
      <![CDATA[Por favor entre en su dirección del email.]]>
    </label>
    -<label id="emailAddressFail">
      <![CDATA[El error: '{emailAddress}' no es una dirección del email válida.]]>
    </label>
    -<label id="emailAddressReEnter">
      <![CDATA[Por favor el re - entre en su dirección del mail.]]>
    </label>
    -<label id="infoTitle">
      <![CDATA[Su Título: ]]>
    </label>
    -<label id="infoStreet">
      <![CDATA[La Dirección callejera:]]>
    </label>
    -<label id="infoCity">
      <![CDATA[La ciudad:]]>
    </label>
    -<label id="infoState">
      <![CDATA[El estado: ]]>
    </label>
    -<label id="enterAgain">
      <![CDATA[Por favor corrija su información y someta de nuevo.]]>
    </label>
    -<label id="InfoEntered">
      <![CDATA[En su información se ha entrado con éxito en el banco de datos.]]>
    </label>
    -<label id="ErrorPage">
      <![CDATA[Una de las tareas falló al intentar construir la página. Ésta es la
      página de error de sistema.]]>
    </label>
    -<label id="LogonPage">
      <![CDATA[Ésta será la página del logon.]]>
    </label>
    -<label id="emailBelongsTo">
      <![CDATA[...en el desarrollo...]]>
    </label>
  </labels>

```

FIG. 9

```
<?xml version="1.0"?>  
  <images>  
    <img id="company_logo" name="us_logo.jpg">  
    <img id="product1_photo" name="us_product1.jpg">  
  </images>
```

1002 1004

FIG. 10

```
<?xml version="1.0"?>  
  <images>  
    <img id="company_logo" name="esp_logo.jpg">  
    <img id="product1_photo" name="esp_product1.jpg">  
  </images>
```

1002 1004

FIG. 11

10020003-12101

File

Edit

View

Go

Favorite

Help

←

→

⊗

📄

🏠

🔍

📁

🖨️

🅐

📧

BackForw...StopRefreshHomeSearchFavoritePrintFontMail

Address

▼

autobytel.com

Master List of Sub-Sites

1204

1206

Edit Sub-Site - Sample Customer\My Garage\Registration\EN-US	Show Site
Edit Sub-Site - Sample Customer\My Garage\Registration\ES	Show Site
Edit Sub-Site - FinConnect\Dealer Tools\HomePage\EN-US	Show Site
Edit Sub-Site - My\New\Site\EN-US	Show Site

1208

1202

New Sub-Site (will be created in c:\inetpub\wwwroot\gb\)

Product:

Program:

Process:

Add Sub-Site

1210

FIG. 12

10000003 121104

File Edit View Go Favorite Help

←

→

⊗

A

Back Forw... Stop Refresh Home Search Favorite Print Font Mail

Address

Site pages for: Sample Customer\My Garage\Registration\ES

Edit Page - Enter email address	Edit Source
Edit Page - Re-enter email address	Edit Source
Edit Page - Email exists	Edit Source
Edit Page - Enter user information	Edit Source
Edit Page - Re-enter user information	Edit Source
Edit Page - Info entered	Edit Source
Edit Page - System error message	Edit Source
Edit Page - Logon Page	Edit Source

New Page Name: Add Page

Site tasks for: Sample Customer\My Garage\Registration\ES

validateEmailAddress	Edit Source
isEmailAddressAvailable	Edit Source
validateUserInformation	Edit Source
createUserInformation	Edit Source
getUserInformation	Edit Source
lookforemail	Edit Source

New Task Name: Add Task

FIG. 13


```

<?xml version="1.0" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
- <xsl:template match="process">
  <xsl:apply-templates />
</xsl:templates>
- <xsl:template match="process">
- <html>
- <head>
- <title>
  <xsl:value-of select="labels/label[@id='title']" disable-output-
    escaping="yes" />
</title>
</head>
- <body>
  <!-- Header template is defined in the included header.xml file -->
  <xsl:call-template name="header" />
  <!-- Label above the email address input box. -->
  <xsl:value-of select="labels/label[@id='emailAddressEnter']" disable-output-
    escaping="yes" />
  - <form name="formEmail" action="default.asp" method="post">
    <!-- Label next to the email address input box. -->
    <xsl:value-of select="labels/label[@id='emailAddress']" disable-output-
      escaping="yes" />
    <input type="text" name="email" />
    <!-- Hidden used for getting to next stage -->
    <input type="hidden" name="page" value="4" />
    <input type="submit" value="Submit" />
  </form>
  <!-- Footer template is defined in the included footer.xml file -->
  <xsl:call-template name="footer" />
  - <br>
    <xsl:value-of select="labels/label[@id='NewLabelName']" disable-output-
      escaping="yes" />
  </br>
</body>
</html>
</xsl:template>
<!-- Include for Header and Footer XSL templates -->
<xsl:include href="header.xml" />
<xsl:include href="footer.xml" />
</xsl:stylesheet>

```

FIG. 15

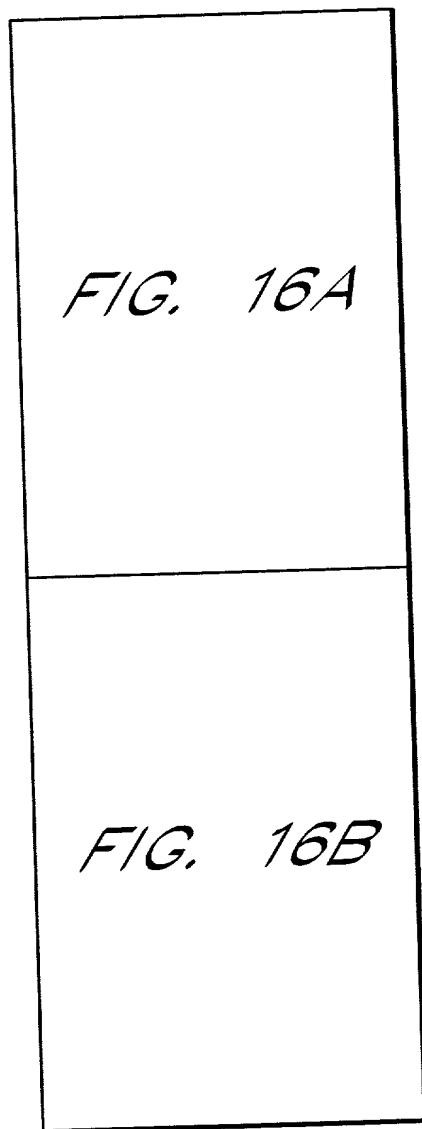


FIG. 16

10020003 161001

File Edit View Go Favorite Help

Back Forw... Stop Refresh Home Search Favorite Print Font Mail

Address

autobytel.com

Edit labels for page: Enter user information ¹⁶⁰²

¹⁶⁰⁴ title Autobytel.com Customer Registration ¹⁶⁰⁶

informationalForm Pleas fill out the form.

emailAddress Email address:

informationRequired REQUIRED INFORMATION:

infoFName First Name:

infoLName Last Name:

infoPass Password:

infoConfirmPass Confirm Password:

Update Labels ¹⁶⁰⁸

¹⁶¹⁰ NewLabelName Enter Label Text Here

¹⁶¹² ¹⁶¹⁴ Add Label ¹⁶¹⁶

FIG. 16A

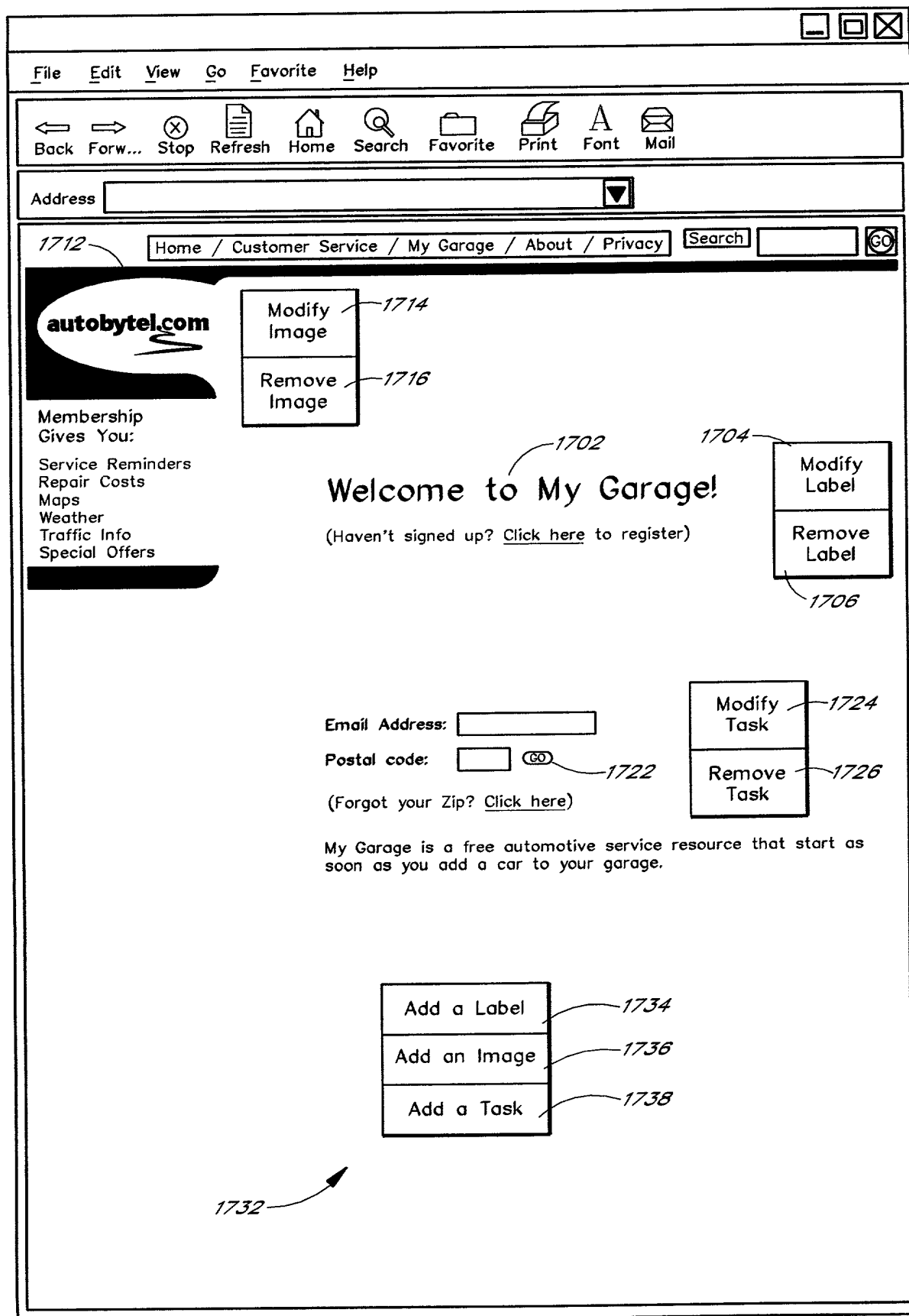


FIG. 17